

《高校教务管理系统》项目开发报告

一、项目概述

基本内容：

1. 项目名称：《高校教务管理系统》
2. 项目背景：教务管理系统在高校教学管理信息化中起着关键的角色，帮助高校提高教学效率、优化资源利用、加强学生管理和教师管理，并提供数据支持和决策依据，促进高校教学质量的提升。同时，教务管理系统也是我们最常使用、最为熟悉的软件之一。因此把高校教务管理系统作为软件工程实训的项目是一个不错的选择。
3. 项目介绍：该系统分为管理员端、教师端和学生端。其中管理员可以管理教师、学生、课程等信息；教师主要负责教学内容管理；学生可以查看与自己有关的信息。

主要参加人员

组 长：XXX

成 员：XXX、XXX

业绩权重说明

均分（1 1 1）

工作任务的分解和人员分工

- XXX：撰写 MySQL 数据库建表代码，Qt 封装数据库操作类，开发系统登录界面、选课界面、公告界面，代码整合优化，界面美化，系统测试。
- XXX：撰写 MySQL 数据库建表代码，开发管理员界面，代码整合优化，界面美化，系统测试。
- XXX：开发教师界面、学生界面，代码整合优化。

创新点和技术难点说明：

1. 创新点：使用了三层架构
2. 技术难点：数据库数据类型繁杂，多种数据之间关系复杂，各个功能模块的数据交互难以正确处理。

二、业务需求分析和基本模块功能

需求分析

随着教育行业的发展和高校规模的不断扩大，高校的学生数量、教师数量和业务量急剧增加。面对着庞大而烦琐的一系列数据管理,传统的人工管理模式将无法满足高校教务管理需求。因此，需要开发一套高校教务管理系统来提高教务管理效率，方便学生和教师获取相关信息，促进学校管理和教学质量的提升。教务管理主要由学生管理、课程管理、成绩管理、教师管理、通知公告等环节组成。教务管理系统要求能够基本实现学生信息管理、课程安排、成绩录入与查看、教师信息管理、通知公告发布等功能。本系统的功能需求如下：

1. 学生管理：包括学生信息录入、修改、查询、删除，学生成绩管理等。
2. 课程管理：包括课程信息录入、修改、查询，分配课程教师等。
3. 成绩管理：包括成绩录入、成绩查询、成绩统计分析等。
4. 教师管理：包括教师信息录入、修改、查询等。
5. 选课管理：包括学生选课、退课，老师筛选学生，淘汰已筛选学生等。
6. 通知公告：包括教务公告发布、院系公告发布、公告查看等。
7. 学生评教：包括学生对老师教学效果打分，分数提交等
8. 权限管理：包括用户角色定义、权限分配、用户登录验证等。
9. 教学质量评估：对教学质量进行评估和统计分析等。
10. 修改个人信息：包括用户查看，修改个人账户信息等。

基本模块功能

1. 系统登录模块
 - (1) 系统登录：管理员、教师和学生通过合法账号和密码登录教务系统。
2. 学生管理模块
 - (1) 学生信息管理：包括学生信息录入、修改、查询、删除。
 - (2) 学生成绩管理：包括学生成绩评判、统计等。

3. 课程管理模块

- (1) 课程信息管理：包括课程信息录入、修改、查询。
- (2) 课程安排：包括课程调整等。

4. 成绩管理模块

- (1) 成绩录入：教师录入学生成绩。
- (2) 成绩查看：学生查看个人成绩。
- (3) 成绩统计分析：对学生成绩进行统计分析。
- (4) 成绩报表生成：生成学生成绩报表。

5. 教师管理模块

- (1) 教师信息管理：包括教师信息录入、修改、查询。
- (2) 教师评价：学生对教师的评价和反馈。

6. 选课管理模块

- (1) 学生选课：学生进行选课操作。
- (2) 退课管理：学生进行退课操作。
- (3) 老师筛选学生：老师对学生进行筛选操作。
- (4) 淘汰已筛选学生：对已筛选学生进行淘汰操作。

7. 通知公告模块

- (1) 通知发布：管理员发布教务公告。
- (2) 公告发布：管理员发布院系公告。
- (3) 通知查询：用户查询系统通知。
- (4) 公告查询：用户查询学校公告。

8. 学生评教模块

- (1) 学生评教：学生对老师的教学效果进行评价和反馈。
- (2) 评教提交：学生提交评教结果。

9. 权限管理模块

- (1) 用户角色定义：定义系统用户的角色和权限。
- (2) 权限分配：分配用户的操作权限。
- (3) 用户登录验证：验证用户的身份和权限。

10. 教学质量评估模块

(1) 教学质量评估：对教学质量进行评估和统计分析。

11. 个人账户信息模块

(1) 用户修改个人信息，包括性别，姓名，年龄等。

三、系统设计与实现

计算机体系设计（采用技术和平台）

计算机体系设计（采用技术和平台）：Qt6.6, MySQL8.0.35, DataGrip2023.2.3.0

功能详细设计与数据库字典

1. 功能详细设计

系统管理员端、教师端、学生端设计类似，下面以管理员端为代表，介绍其功能详细设计。

1、系统登录

管理员、教师和学生通过合法账号、密码、身份登录教务系统，系统将验证登录用户的账号、密码和身份信息是否正确，并给出相应的提示。如果验证通过，则进入身份对应的主界面；验证失败，则弹出提示窗口。

2、学生管理

管理员界面中的学生管理窗口可以显示所有存在的学生账户信息，在学生管理窗口中有添加账户、编辑账户、删除账户、查找账户 4 个功能按钮。当添加账户按钮（需要选中一个学生账户）、编辑账户按钮（需要选中一个学生账户）、查找账户按钮（需要在输入框中输入要查找学生的学号）被点击时，系统会打开同一个管理学生账户的对话框，管理员可以点击对话框中的编辑按钮编辑账户的信息，点击对话框中的保存按钮保存账户信息，点击取消按钮取消对账户的操作。管理员点击删除按钮后，系统会弹出一个提示框，确认是否删除被选中的学生账户，选择“是”，系统会删除该学生的账户，选择“否”，系统不会做任何操作。

3、课程管理

管理员界面中的课程管理窗口可以显示所有存在的课程信息，在课程管理窗口中有添加课程、编辑课程、删除课程、查找课程 4 个功能按钮。当添加课程按钮（需要选中一个课程）、编辑课程按钮（需要选中一个课程）、查找课程按钮（需要在输入框中输入要查找课程的编号）被点击时，系统会打开同一个管理课程的对话框，管理员可以点击对话框中的编辑按钮编辑课程的信息，点击对话框中的保存按钮保存课程信息，点击取消按钮取消对课程信息的操作。管理员点击

删除按钮后，系统会弹出一个提示框，确认是否删除被选中的课程，选择“是”，系统会删除该课程，选择“否”，系统不会做任何操作。

4、成绩管理

在管理员界面，该模块负责评判学生是否通过考试（60 分及以上），并且如果有学生作弊被举报，该生的通过状态还可以手动更改。在教师界面，该模块可以自动统计并分析各个分数层的人数及占比，并且通过绘制图表展示结果。学生界面，该模块可以自动统计分析各个分数层的科目数量以及占比，并且通过绘制图表呈现结果。

5、教师管理

管理员界面中的教师管理窗口可以显示所有存在的教师账户信息，在教师管理窗口中有添加账户、编辑账户、删除账户、查找账户 4 个功能按钮。当添加账户按钮（需要选中一个教师账户）、编辑账户按钮（需要选中一个教师账户）、查找账户按钮（需要在输入框中输入要查找教师的职工号）被点击时，系统会打开同一个管理教师账户的对话框，管理员可以点击对话框中的编辑按钮编辑账户的信息，点击对话框中的保存按钮保存账户信息，点击取消按钮取消对账户的操作。管理员点击删除按钮后，系统会弹出一个提示框，确认是否删除被选中的教师账户，选择“是”，系统会删除该教师的账户，选择“否”，系统不会做任何操作。

6、选课管理

学生界面，该模块分为可选课程和已选课程两部分，并且可选课程部分提供公选，公必，专选，专必四种课程展示。学生选择某门课程后，相应的课程会从可选列表移入已选列表，并展示当前选课状态（待筛选、选课成功）。在已选课程部分学生可以选择退课，相应课程会转移至可选课程列表中。

在教师界面，该模块分为待筛选学生和已选学生两部分，由老师从待筛选列表中选择学生移入已选列表，或从已选列表中淘汰学生转入待筛选列表。教师界面只展示老师负责课程的选课列表，管理员界面展示所有课程的选课列表，并且管理员可以任意选择、淘汰学生。

7、通知公告管理

该模块分为院系公告和教务公告两部分。管理员可以任意删除，修改，发布公告，同时该模块支持双击某条公告，跳出新的界面展示该公告的具体内容。（学生、教师界面只能查看公告）

8、学生评教

在管理员界面，该模块负责查看学生的评教结果。管理员可以通过教师编号搜索并查看教师的评教分数。在学生界面，该模块负责评判老师的教学质量。学生可以在教学态度、教学内容、教学方法、教学效果四个地方打分，点击提交即可保存评教结果。最后系统会自动生成总分保存在总分一栏。在教师界面，老师

只能看到评教结果。

9、信息修改

管理员在该界面可以修改个人信息， 点击编辑按钮即可对个人信息进行修改， 其中账号、姓名、职称不可修改， 其他则需要按标准输入， 点击保存按钮即可保存信息。老师， 学生同理。

2. 数据库字典

数据库的设计在软件工程中占据着举足轻重的地位。数据库如果设计不合理、不完善， 将在系统开发过程中， 甚至到后期的系统维护、功能变更和功能扩大时引起较多的问题， 甚至需要重新设计， 重做大量已完成的工作。一个成功的数据库设计会给软件的开发过程及后期的维护带来极大的益处。

〔1〕 院系数据表

院系数据表用来存储院系的基本信息， 包括院系编号、院系名称。院系数据表如表 3-1 所示。

表 3-1 院系数据表

字段名	数据类型	NULL	键	默认值	额外信息
departmentId (院系编号)	int	否	主键	NULL	自增长
departmentName (院系名称)	nvarchar(40)	否	—	NULL	—

〔2〕 专业数据表

专业数据表用来存储专业信息， 包括专业编号、专业所属院系的编号、专业名称、专业毕业学分。专业数据表如表 3-2 所示。

表 3-2 专业数据表

字段名	数据类型	NULL	键	默认值	额外信息
majorId (专业编号)	int	否	主键	NULL	自增长

departmentId (所属院系编号)	int	是	—	NULL	—
majorName (专业名称)	nvarchar(40)	否	—	NULL	—
majorSum (毕业总学分)	int	是	—	NULL	—

〔3〕 公告数据表

公告数据表用来存储公告的信息，包括公告编号、公告类型、院系编号（当公告属于院系公告时）、公告内容、公告联系人、联系人电话、联系人邮箱、公告发布日期。公告数据表如表 3-3 所示。

表 3-3 公告数据表

字段名	数据类型	NULL	主键	默认值	额外信息
announceId (公告编号)	int	否	主键	NULL	自增长
announceType (公告类型)	int	否	—	NULL	announceType >= 0 AND announceType <= 1
departmentId (院系编号)	int	是	—	NULL	—
announceContent (公告内容)	text	是	—	NULL	—
announceLinkman (公	nvarchar(8)	是	—	NULL	—

告联系人)					
announcePhone (联系人电话)	nvarchar(20)	是	—	NULL	—
announceMailbox (联系人邮箱)	nvarchar(30)	是	—	NULL	—
announceDate (公告发布日期)	date	是	—	CURRENT_DATE()	DEFAULT_GENERATED

〔4〕 管理员数据表

管理员数据表用来存储管理员的信息，包括管理员编号、管理员登陆密码、管理员姓名、管理员性别、管理员年龄、管理员电话、管理员邮箱。管理员数据表如表 3-4 所示。

表 3-4 管理员数据表

字段名	数据类型	NULL	键	默认值	额外信息
adminId (管理员编号)	int	否	主键	10000000	自增长
adminPassword (管理员登陆密码)	nvarchar(20)	否	—	NULL	—
adminName (管理员姓名)	nvarchar(8)	否	—	NULL	—
adminSex (管理员性别)	varchar(2)	否	—	NULL	CHECK (adminSex IN ('男', '女'))

adminAge (管理员年龄)	int	否	—	NULL	—
adminPhoneNum (管理员电话)	nvarchar(20)	否	—	NULL	—
adminMailbox (管理员邮箱)	nvarchar(30)	是	—	NULL	—

〔5〕 教师数据表

教师数据表用来存储教师的信息，包括教师编号、教师登陆密码、教师姓名、教师性别、教师年龄、所属院系的编号、职称、教师电话、教师邮箱、教师简介。教师数据表如表 3-5 所示。

表 3-5 管理员数据表

字段名	数据类型	NULL	键	默认值	额外信息
teacherId (教师编号)	int	否	主键	20000000	自增长
teacherPassword (教师登陆密码)	nvarchar(20)	否	—	NULL	—
teacherName (教师姓名)	nvarchar(8)	否	—	NULL	—
teacherSex (教师性别)	nvarchar(2)	否	—	NULL	CHECK(teacherSex IN ('男','女'))
teacherAge (教师年	int	否	—	NULL	—

龄)					
department Id (所属院系 的编号)	int	是	—	NULL	—
teacherJob Title (职 称)	nvarchar(8)	否	—	NULL	—
teacherPho neNum (教 师电话)	nvarchar(2 0)	否	—	NULL	—
teacherMail box (教师 邮箱)	nvarchar(3 0)	是	—	NULL	—
teacherIntr oduction (教师简 介)	text	是	—	NULL	—

[6] 学生数据表

学生数据表用来存储学生的信息，包括学生编号、学生所属专业、学生所属院系、学生登陆密码、学生姓名、学生性别、学生年龄、毕业总学分。学生数据表如表 3-6 所示。

表 3-6 学生数据表

字段名	数据类型	NULL	键	默认值	额外信息
studentId (学生编 号)	int	否	主键	30000000	自增长
majorId (学生所属 专业)	int	是	—	NULL	—

departmentId (学生所属院系)	int	是	—	NULL	—
studentPassword (学生登陆密码)	nvarchar(20)	否	—	NULL	—
studentName (学生姓名)	nvarchar(8)	否	—	NULL	—
studentSex (学生性别)	nvarchar(2)	否	—	NULL	CHECK(studentSex IN ('男', '女'))
studentAge (学生年龄)	int	是	—	NULL	—
studentSumScore (毕业总学分)	int	是	—	0	—

〔7〕 课程数据表

课程数据表用来存储课程的信息，包括课程编号、授课老师编号、开课专业编号、开课院系编号、课程学期、课程类型、学时、学分、开课时间、结课时间、授课老师姓名、课程名称、上课时间、上课地点、课程可选人数、课程已选人数。课程数据表如表 3-7 所示。

表 3-7 课程数据表

字段名	数据类型	NULL	键	默认值	额外信息
courseId (课程编号)	int	否	主键	NULL	自增长

teacherId (授课老师 编号)	int	是	外键	NULL	FOREIGN KEY (teacherId) REFEREN CES teacher_ta ble(teacher Id)
majorId (开课专业 编号)	int	是	—	NULL	—
department Id (开课院 系编号)	int	是	—	NULL	—
courseTer m (课程学 期)	int	是	—	NULL	CHECK(co urseTerm > = 1 AND courseTer m <= 2)
courseType (课程类 型)	int	是	—	NULL	—
courseHour (学时)	int	是	—	NULL	—
courseScor e (学分)	float	是	—	NULL	—
courseBegi nDate (开 课时间)	date	是	—	NULL	—
courseEnd Date (结课 时间)	date	是	—	NULL	—

teacherName (授课老师姓名)	nvarchar(20)	是	—	NULL	—
courseName (课程姓名)	nvarchar(40)	是	—	NULL	—
courseClassTime (上课时间)	text	是	—	NULL	—
placeOfClass (上课地点)	nvarchar(100)	是	—	NULL	—
courseMaxCount (课程可选人数)	int	是	—	NULL	—
courseSelectNum (课程已选人数)	int	是	—	NULL	—

〔8〕 评教数据表

评教数据表用来存储评教的信息，包括评教记录编号、评教学生编号、课程编号、教师编号、课程学期、专业编号、院系编号、课程名称、教学态度得分、教学内容得分、教学方法得分、教学效果得分、建议、总得分。评教数据表如表 3-8 所示。

表 3-8 评教数据表

字段名	数据类型	NULL	键	默认值	额外信息
evaluateId (评教记录编号)	int	否	主键	NULL	自增长
studentId	int	是	—	NULL	—

(评教学生 编号)					
courseId (课程编 号)	int	是	—	NULL	—
teacherId (教师编 号)	int	是	—	NULL	—
courseTer m (课程学 期)	int	是	—	NULL	—
majorId (专业编 号)	int	是	—	NULL	—
department Id (院系编 号)	int	是	—	NULL	—
courseNam e (课程名 称)	nvarchar(2 0)	是	—	NULL	—
evaluateSt atus (教学 态度得分)	int	是	—	NULL	CHECK (evaluateSt atus <= 25)
evaluateCo ntent (教学 内容得分)	int	是	—	NULL	CHECK (evaluateC ontent <= 30)
evaluateMe thod (教学 方法得分)	int	是	—	NULL	CHECK (evaluateM ethod <= 30)

evaluateEffect (教学效果得分)	int	是	—	NULL	CHECK (evaluateEffect <= 15)
evaluateSuggest (建议)	text	是	—	NULL	—
sumScore (上总得分)	int	是	—	NULL	—

[9] 选课数据表

选课数据表用来存储选课的信息，包括选课记录编号、课程编号、教师编号、学生编号、专业编号、院系编号、课程学期、选课状态、课程名称。选课数据表如表 3-9 所示。

表 3-9 选课数据表

字段名	数据类型	NULL	键	默认值	额外信息
selectId (选课记录编号)	int	否	主键	NULL	自增长
courseId (课程编号)	int	否	外键	NULL	FOREIGN KEY (courseId) REFERENCES course_table(courseId)
teacherId (教师编号)	int	否	外键	NULL	FOREIGN KEY (teacherId) REFERENCES teacher_table(teacher

					Id)
studentId (学生编号)	int	否	外键	NULL	FOREIGN KEY (studentId) REFEREN CES student_tab le(studentId)
majorId (专业编号)	int	是	—	NULL	—
department Id (院系编号)	int	是	—	NULL	—
courseTerm (课程学期)	int	否	—	NULL	CHECK (courseTerm >= 1 AND courseTerm <= 2)
selectStatus (选课状态)	int	是	—	NULL	CHECK (selectStatus >= -1 AND selectStatus <= 2)
courseName (课程名称)	nvarchar(40)	是	—	NULL	—

〔10〕成绩数据表

成绩数据表用来存储成绩的信息，包括成绩记录编号、课程编号、学生编号、教师编号、课程学期、成绩、通过状态、补考次数。成绩数据表如表 3-10 所示。

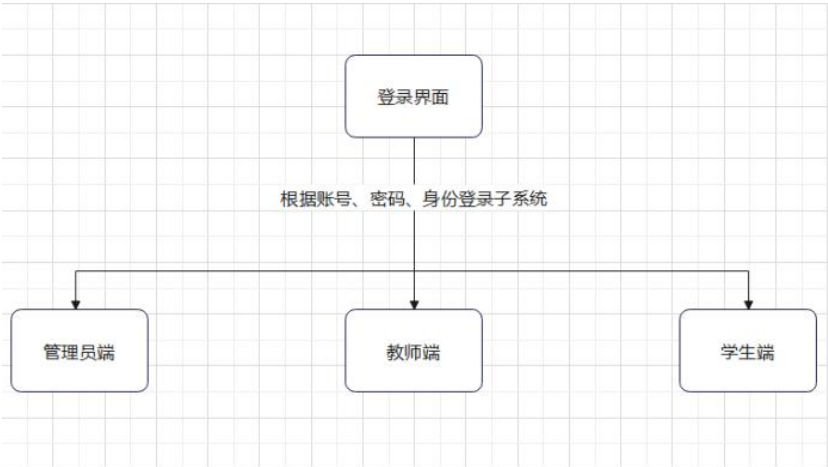
表 3-10 成绩数据表

字段名	数据类型	NULL	键	默认值	额外信息
scoreId (成绩记录 编号)	int	否	主键	NULL	自增长
courseId (课程编 号)	int	否	外键	NULL	FOREIGN KEY (courseId) REFEREN CES course_tabl e(courseId)
studentId (学生编 号)	int	否	外键	NULL	FOREIGN KEY (teacherId) REFEREN CES teacher_ta ble(teacher Id)
teacherId (教师编 号)	int	否	外键	NULL	FOREIGN KEY (studentId) REFEREN CES student_tab le(studentId)
courseTer m (课程学 期)	int	是	—	NULL	CHECK (courseTer m >= 1 AND courseTer m <= 2)
score (成 绩)	float	是	—	NULL	—

studentPass (通过状态)	int	是	—	NULL	—
studentRetryNum (补考次数)	int	是	—	NULL	—

对象抽象和类层次图

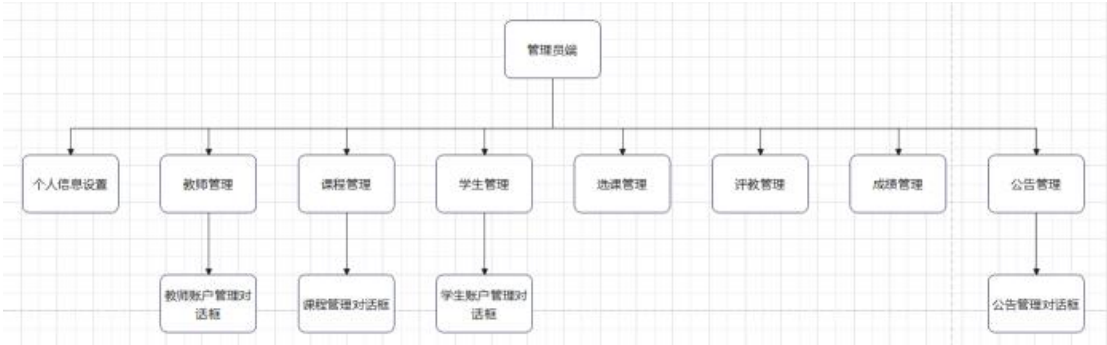
登陆界面层次图



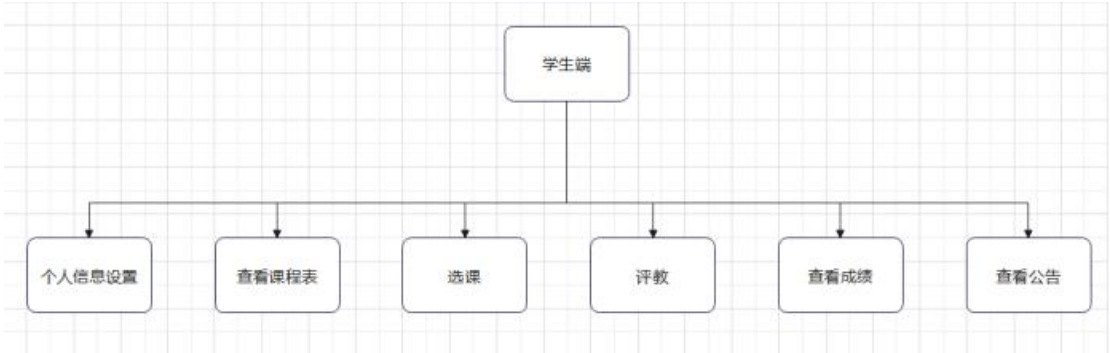
数据库层次图



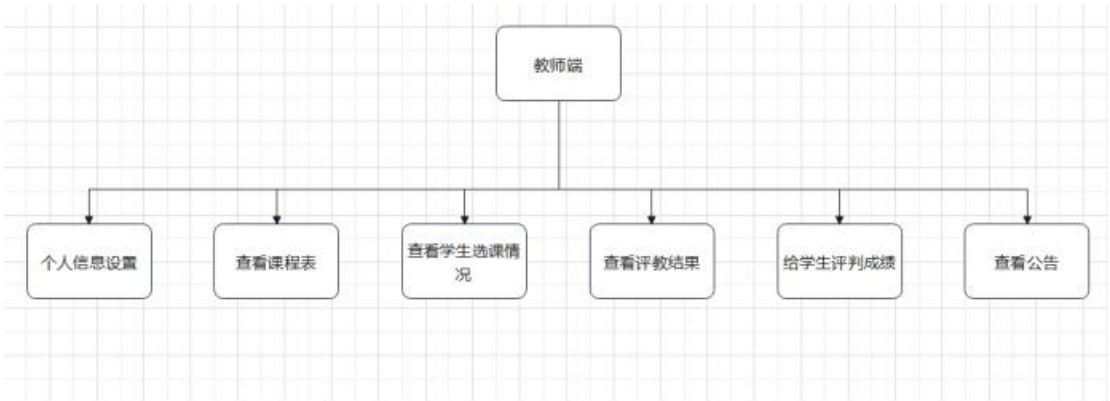
管理员类层次图



学生类层次图



教师类层次图



编码实现（举一个有代表性的功能实现）

代表性功能：管理端的课程管理



课程管理界面

课程管理对话框

×

教师编号	<input type="text"/>	教师姓名	<input type="text"/>
课程编号	9	课程名称	<input type="text"/>
课程学期	<input type="text"/>	课程类型	公必
课程学分	<input type="text"/>	课 时	<input type="text"/>
开课时间	2000-01-01	结课时间	2000-01-01
上课时间	<input type="text"/>	上课地点	<input type="text"/>
课程人数	<input type="text"/>	已选人数	<input type="text"/>
专 业	计算机科学与技术	院 系	计算机科学与技术学院
<div>编辑</div>		<div>保存</div> <div>取消</div>	

课程管理对话框

```
C++
// 课程管理界面头文件 coursemanagementwindow.h
#ifndef COURSEMANAGEMENTWINDOW_H
#define COURSEMANAGEMENTWINDOW_H

#include <QWidget>
#include "sqlmodelall.h"
#include "admindialog3.h"
#include <QMessageBox>

namespace Ui {
class CourseManagementWindow;
}

class CourseManagementWindow : public QWidget
{
    Q_OBJECT

public:
```

```

        explicit CourseManagementWindow(QWidget *parent = nullptr);
        ~CourseManagementWindow();

private slots:

    void on_pushButtonAdd_clicked();

    void on_pushButtonEdit_clicked();

    void on_pushButtonDelete_clicked();

    void on_pushButtonSearch_clicked();

private:
    Ui::CourseManagementWindow *ui;

    AdminDialog3* dialog3;
    //QSqlTableModel* model;
    QSqlRelationalTableModel *model;
    QItemSelectionModel* selectionModel;

    CourseTableSql csql;
};

#endif // COURSEMANAGEMENTWINDOW_H

```

```

C++
// 课程管理界面源文件 coursemanagementwindow.cpp
#include "coursemanagementwindow.h"
#include "ui_coursemanagementwindow.h"

#include <QDebug>
CourseManagementWindow::CourseManagementWindow(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::CourseManagementWindow)
{
    ui->setupUi(this);
    ui->lineEditSearch->setText("请输入课程编号");

    model = csql.getRelationModel(this);

    //设置关系

```

```

        MajorSql major; DepartmentSql depart;
        QString majorTableName=major.getTableName();
        QString dapartTableName=depart.getTableName();
        model-
>setRelation(2, QSqlRelation(majorTableName, "majorId", "majorName"))
;
        model-
>setRelation(3, QSqlRelation(dapartTableName, "departmentId", "depart
mentName"));

        ui->tableView->setModel(model);
        model->select();
        ui->tableView-
>setEditTriggers(QAbstractItemView::NoEditTriggers);//    设置为只
读
        ui->tableView-
>setSelectionMode(QAbstractItemView::SingleSelection);//    单行选中
        ui->tableView-
>setSelectionBehavior(QAbstractItemView::SelectRows);

        //隐藏课程类型
        ui->tableView->setColumnHidden(5,1);
        selectionModel = ui->tableView->selectionModel();

        dialog3 = new AdminDialog3(this);
    }

CourseManagementWindow::~CourseManagementWindow()
{
    delete ui;
}

void CourseManagementWindow::on_pushButtonAdd_clicked()
{
    int newId, rowCount = model->rowCount();
    if (rowCount == 0)
        newId = 1;
    else {
        QSqlRecord record = model->record(rowCount - 1);
        newId = record.value("courseId").toInt() + 1;
    }
    dialog3->initDialog1(newId, model);
    dialog3->open();
}

```

```

void CourseManagementWindow::on_pushButtonEdit_clicked()
{
    int curRow = selectionModel->currentIndex().row();
    if (curRow != -1) {
        QSqlRecord record = model->record(curRow);
        int id = record.value("courseId").toInt();
        qDebug() << id;
        dialog3->initDialog2(id, model);
        dialog3->open();
    }
}

void CourseManagementWindow::on_pushButtonDelete_clicked()
{
    int curRow = selectionModel->currentIndex().row();
    if (curRow != -1) {
        QMessageBox::StandardButton reply;
        reply = QMessageBox::question(this, "确认删除", "确定要删除
选中的行吗?",

QMessageBox::Yes|QMessageBox::No);
        if (reply == QMessageBox::Yes) {

            // 删除选中的行
            CourseTableSql courseList;
            int id = model->data(model-
>index(curRow,0),Qt::EditRole).toInt();
            courseList.deleteSingleRecord(id);

            // 提交变更到数据库
            model->submitAll();

            // 刷新视图
            model->select();
        }
    }
}

void CourseManagementWindow::on_pushButtonSearch_clicked()
{
    QSqlRecord record = csq1.findSingleRecord(ui->lineEditSearch-
>text().toInt());

```

```

        if (record.isNull("courseId")) {
            QMessageBox::information(this, "提示", "课程不存在");
            return;
        }
        dialog3->initDialog2(record.value("courseId").toInt(), model);
        dialog3->open();
    }
}

```

C++

```

// 课程管理界面 ui 设计
#ifndef UI_COURSEMANAGEMENTWINDOW_H
#define UI_COURSEMANAGEMENTWINDOW_H

#include <QtCore/QVariant>
#include <QtWidgets/QApplication>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QHeaderView>
#include <QtWidgets/QLineEdit>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QTableView>
#include <QtWidgets/QWidget>

QT_BEGIN_NAMESPACE

class Ui_CourseManagementWindow
{
public:
    QGridLayout *gridLayout;
    QHBoxLayout *horizontalLayout;
    QPushButton *pushButtonAdd;
    QPushButton *pushButtonEdit;
    QPushButton *pushButtonDelete;
    QSpacerItem *horizontalSpacer;
    QLineEdit *lineEditSearch;
    QPushButton *pushButtonSearch;
    QTableView *tableView;

    void setupUi(QWidget *CourseManagementWindow)
    {
        if (CourseManagementWindow->objectName().isEmpty())
            CourseManagementWindow->

```



```
>setObjectName("CourseManagementWindow");
    CourseManagementWindow->resize(567, 444);
    gridLayout = new QGridLayout(CourseManagementWindow);
    gridLayout->setObjectName("gridLayout");
    horizontalLayout = new QHBoxLayout();
    horizontalLayout->setObjectName("horizontalLayout");
    pushButtonAdd = new QPushButton(CourseManagementWindow);
    pushButtonAdd->setObjectName("pushButtonAdd");

    horizontalLayout->addWidget(pushButtonAdd);

    pushButtonEdit = new QPushButton(CourseManagementWindow);
    pushButtonEdit->setObjectName("pushButtonEdit");

    horizontalLayout->addWidget(pushButtonEdit);

    pushButtonDelete = new
QPushButton(CourseManagementWindow);
    pushButtonDelete->setObjectName("pushButtonDelete");

    horizontalLayout->addWidget(pushButtonDelete);

    horizontalSpacer = new QSpacerItem(48, 20,
QSizePolicy::Expanding, QSizePolicy::Minimum);

    horizontalLayout->addItem(horizontalSpacer);

    lineEditSearch = new QLineEdit(CourseManagementWindow);
    lineEditSearch->setObjectName("lineEditSearch");

    horizontalLayout->addWidget(lineEditSearch);

    pushButtonSearch = new
QPushButton(CourseManagementWindow);
    pushButtonSearch->setObjectName("pushButtonSearch");

    horizontalLayout->addWidget(pushButtonSearch);

    gridLayout->addLayout(horizontalLayout, 0, 0, 1, 1);

    tableView = new QTableView(CourseManagementWindow);
    tableView->setObjectName("tableView");
```

```

        gridLayout->addWidget(tableView, 1, 0, 1, 1);

        retranslateUi(CourseManagementWindow);

        QMetaObject::connectSlotsByName(CourseManagementWindow);
    } // setupUi

    void retranslateUi(QWidget *CourseManagementWindow)
    {
        CourseManagementWindow-
>setWindowTitle(QCoreApplication::translate("CourseManagementWindo
w",
"\350\257\276\347\250\213\347\256\241\347\220\206\347\252\227\345\
217\243", nullptr));
        pushButtonAdd-
>setText(QCoreApplication::translate("CourseManagementWindow",
"\346\267\273\345\212\240\350\257\276\347\250\213", nullptr));
        pushButtonEdit-
>setText(QCoreApplication::translate("CourseManagementWindow",
"\344\277\256\346\224\271\350\257\276\347\250\213", nullptr));
        pushButtonDelete-
>setText(QCoreApplication::translate("CourseManagementWindow",
"\345\210\240\351\231\244\350\257\276\347\250\213", nullptr));
        pushButtonSearch-
>setText(QCoreApplication::translate("CourseManagementWindow",
"\346\237\245\346\211\276\350\257\276\347\250\213", nullptr));
    } // retranslateUi

};

namespace Ui {
    class CourseManagementWindow: public Ui_CourseManagementWindow
    {};
} // namespace Ui

QT_END_NAMESPACE

#endif // UI_COURSEMANAGEMENTWINDOW_H

```

```

C++
// 课程管理对话框头文件 admindialog3.h
#ifndef ADMINDIALOG3_H

```

```

#define ADMINDIALOG3_H

#include <QDialog>
#include <QMap>
#include "sqlmodelall.h"
#include <QMessageBox>

namespace Ui {
class AdminDialog3;
}

class AdminDialog3 : public QDialog
{
    Q_OBJECT

public:
    explicit AdminDialog3(QWidget *parent = nullptr);
    ~AdminDialog3();
    //    bool isValid();
    void initDialog1(int id, QSqlTableModel* m);
    void initDialog2(int id, QSqlTableModel* m);

private slots:
    void on_pushButtonEdit_clicked();

    void on_pushButtonSave_clicked();

    void on_comboBoxDepartment_currentIndexChanged(int index);

    void on_lineEditTeacherId_returnPressed();

private:
    Ui::AdminDialog3 *ui;
    int courseId;
    CourseTableSql csq;
    QSqlTableModel* model;
    QList<QString> nameList;
    QMap<QString,QString> mp;
};

#endif // ADMINDIALOG3_H

```

C++

```

// 课程管理对话框源文件 adminDialog3.cpp
#include "adminDialog3.h"
#include "ui_adminDialog3.h"

#include <QCompleter>
#include <QIntValidator>
#include <QDoubleValidator>

#include <QDebug>

AdminDialog3::AdminDialog3(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::AdminDialog3)
{
    ui->setupUi(this);

    // 设置只读

    nameList<<"公必"<<"公选"<<"专必"<<"专选";
    ui->comboBoxCourseType->addItem(nameList);
    ui->lineEditCourseId->setReadOnly(true);

    //设置三个只能输入数字
    QIntValidator *termValite=new QIntValidator(1,2,this);//学期
    ui->lineEditCourseTerm->setValidator(termValite);

    QIntValidator *hourValite=new QIntValidator(10,200,this);//学
    时
    ui->lineEditCourseHour->setValidator(hourValite);

    QIntValidator *personValite=new QIntValidator(1,9000,this);
    ui->lineEditHeadCount->setValidator(personValite);

    QIntValidator *peopleValite=new QIntValidator(1,9000,this);
    ui->lineEditSelectNum->setValidator(peopleValite);

    QDoubleValidator *scoreValite = new QDoubleValidator(0.5, 5,
    1, this); // 学分
    scoreValite->setNotation(QDoubleValidator::StandardNotation);
    // 设置为标准表示法
    ui->lineEditCourseScore->setValidator(scoreValite);

```

```

TeacherSql teacherList;
QSqlTableModel *teacherModel=teacherList.getTableModel(this);
QStringList idNames;
for(int i=0;i<teacherModel->rowCount();++i)
{
    QModelIndex index=teacherModel->index(i,0);
    QModelIndex nameIndex=teacherModel->index(i,teacherModel-
>fieldIndex("teacherName"));
    idNames<<teacherModel-
>data(index,Qt::EditRole).toString();
    mp.insert(teacherModel-
>data(index,Qt::EditRole).toString(),teacherModel-
>data(nameIndex,Qt::EditRole).toString());
}

QCompleter *compelter=new QCompleter(idNames,this);
ui->lineEditTeacherId->setCompleter(compelter);

//读取专业
MajorSql major;

QSqlTableModel *majorModel=major.getTableModel(this);
for(int i=0;i<majorModel->rowCount();++i)
{
    QModelIndex index=majorModel->index(i,majorModel-
>fieldIndex("majorName"));
    QString name = majorModel-
>data(index,Qt::EditRole).toString();
    ui->comboBoxMajor->addItem(name);
}

//读取院系
DepartmentSql depart;
QSqlTableModel *departModel=depart.getTableModel(this);
for(int i=0;i<departModel->rowCount();++i)
{
    QModelIndex index=departModel->index(i,departModel-
>fieldIndex("departmentName"));
    QString name = departModel-
>data(index,Qt::EditRole).toString();
    ui->comboBoxDepartment->addItem(name);
}
}

```

```

AdminDialog3::~AdminDialog3()
{
    delete ui;
}

void AdminDialog3::initDialog1(int id, QSqlTableModel *m)
{
    ui->lineEditTeacherId->clear();
    ui->lineEditTeacherName->clear();
    ui->lineEditCourseId->clear();
    ui->lineEditCourseName->clear();
    ui->lineEditCourseTerm->clear();
    ui->lineEditCourseScore->clear();
    ui->lineEditClassTime->clear();
    ui->lineEditClassLocal->clear();
    ui->lineEditHeadCount->clear();
    ui->lineEditSelectNum->clear();
    ui->lineEditCourseHour->clear();
    ui->dateEditBeginDate->setEnabled(false);
    ui->dateEditEndDate->setEnabled(false);

    ui->lineEditTeacherId->setReadOnly(true);
    ui->lineEditTeacherName->setReadOnly(true);
    ui->lineEditCourseId->setReadOnly(true);
    ui->lineEditCourseName->setReadOnly(true);
    ui->lineEditCourseTerm->setReadOnly(true);
    ui->lineEditCourseScore->setReadOnly(true);
    ui->lineEditClassTime->setReadOnly(true);
    ui->lineEditClassLocal->setReadOnly(true);
    ui->lineEditHeadCount->setReadOnly(true);
    ui->lineEditSelectNum->setReadOnly(true);
    ui->lineEditCourseHour->setReadOnly(true);

    ui->comboBoxCouseType->clear();
    ui->comboBoxCouseType->setEnabled(false);
    ui->comboBoxCouseType->addItem(nameList);

    ui->comboBoxDepartment->clear();
    ui->comboBoxMajor->clear();
    ui->comboBoxDepartment->setEnabled(false);
    ui->comboBoxMajor->setEnabled(false);
}

```

```

//读取专业
MajorSql major;

QSqlTableModel *majorModel=major.getTableModel(this);
for(int i=0;i<majorModel->rowCount();++i)
{
    QModelIndex index=majorModel->index(i,majorModel-
>fieldIndex("majorName"));
    QString name = majorModel-
>data(index,Qt::EditRole).toString();
    ui->comboBoxMajor->addItem(name);
}

//读取院系
DepartmentSql depart;
QSqlTableModel *departModel=depart.getTableModel(this);
for(int i=0;i<departModel->rowCount();++i)
{
    QModelIndex index=departModel->index(i,departModel-
>fieldIndex("departmentName"));
    QString name = departModel-
>data(index,Qt::EditRole).toString();
    ui->comboBoxDepartment->addItem(name);
}

ui->lineEditCourseId->setText(QString("%1").arg(id));
courseId = id;
model = m;
}

void AdminDialog3::initDialog2(int id, QSqlTableModel *m)
{
    ui->lineEditTeacherId->clear();
    ui->lineEditTeacherName->clear();
    ui->lineEditCourseId->clear();
    ui->lineEditCourseName->clear();
    ui->lineEditCourseTerm->clear();
    ui->lineEditClassTime->clear();
    ui->lineEditClassLocal->clear();
    ui->lineEditHeadCount->clear();
    ui->lineEditSelectNum->clear();
    ui->lineEditCourseHour->clear();
}

```

```

ui->lineEditTeacherId->setReadOnly(true);
ui->lineEditTeacherName->setReadOnly(true);
ui->lineEditCourseId->setReadOnly(true);
ui->lineEditCourseName->setReadOnly(true);
ui->lineEditCourseTerm->setReadOnly(true);
ui->lineEditCourseScore->setReadOnly(true);
ui->lineEditClassTime->setReadOnly(true);
ui->lineEditClassLocal->setReadOnly(true);
ui->lineEditHeadCount->setReadOnly(true);
ui->lineEditSelectNum->setReadOnly(true);
ui->lineEditCourseHour->setReadOnly(true);
ui->dateEditBeginDate->setEnabled(false);
ui->dateEditEndDate->setEnabled(false);

ui->comboBoxCouseType->clear();
ui->comboBoxCouseType->addItem(nameList);

ui->comboBoxDepartment->clear();
ui->comboBoxMajor->clear();

//读取专业
MajorSql major;

QSqlTableModel *majorModel=major.getTableModel(this);
for(int i=0;i<majorModel->rowCount();++i)
{
    QModelIndex index=majorModel->index(i,majorModel-
>fieldIndex("majorName"));
    QString name = majorModel-
>data(index,Qt::EditRole).toString();
    ui->comboBoxMajor->addItem(name);
}

//读取院系
DepartmentSql depart;
QSqlTableModel *departModel=depart.getTableModel(this);
for(int i=0;i<departModel->rowCount();++i)
{
    QModelIndex index=departModel->index(i,departModel-
>fieldIndex("departmentName"));
    QString name = departModel-
>data(index,Qt::EditRole).toString();
    ui->comboBoxDepartment->addItem(name);
}

```



```

    }

    courseId = id;
    model = m;
    QSqlRecord record = csql.findSingleRecord(id);

    ui->lineEditTeacherId-
>setText(record.value("teacherId").toString());
    ui->lineEditTeacherName-
>setText(record.value("teacherName").toString());
    ui->lineEditCourseId-
>setText(record.value("courseId").toString());
    ui->lineEditCourseName-
>setText(record.value("courseName").toString());
    ui->lineEditCourseTerm-
>setText(record.value("courseTerm").toString());
    ui->lineEditCourseScore-
>setText(record.value("courseScore").toString());
    ui->dateEditBeginDate-
>setDate(QDate::fromString(record.value("courseBeginDate").toString(
g(),"yyyy-MM-dd")));
    ui->dateEditEndDate-
>setDate(QDate::fromString(record.value("courseEndDate").toString(
),"yyyy-MM-dd"));
    ui->lineEditClassTime-
>setText(record.value("courseClassTime").toString());
    ui->lineEditClassLocal-
>setText(record.value("placeOfClass").toString());
    ui->lineEditHeadCount-
>setText(record.value("courseMaxCount").toString());
    ui->lineEditSelectNum-
>setText(record.value("courseSelectNum").toString());
    ui->lineEditCourseHour-
>setText(record.value("courseHour").toString());

    ui->comboBoxCouseType-
>setCurrentIndex(record.value("courseType").toInt());
    ui->comboBoxDepartment-
>setCurrentIndex(record.value("departmentId").toInt()-1);

    int majorId = record.value("majorId").toInt();
    QSqlRecord majorRecord = major.findSingleRecord(majorId);

```

```

        ui->comboBoxMajor->
>setCurrentText(majorRecord.value("majorName").toString());

        ui->comboBoxDepartment->setEnabled(false);
        ui->comboBoxMajor->setEnabled(false);
        ui->comboBoxCouseType->setEnabled(false);
    }

void AdminDialog3::on_pushButtonEdit_clicked()
{
    ui->lineEditTeacherId->setReadOnly(false);
    ui->lineEditTeacherName->setReadOnly(false);
    ui->lineEditCourseName->setReadOnly(false);
    ui->lineEditCourseTerm->setReadOnly(false);
    ui->lineEditCourseScore->setReadOnly(false);
    ui->lineEditClassTime->setReadOnly(false);
    ui->lineEditClassLocal->setReadOnly(false);
    ui->lineEditHeadCount->setReadOnly(false);
    ui->lineEditSelectNum->setReadOnly(false);
    ui->lineEditCourseHour->setReadOnly(false);
    ui->dateEditBeginDate->setEnabled(true);
    ui->dateEditEndDate->setEnabled(true);
    ui->comboBoxDepartment->setEnabled(true);
    ui->comboBoxMajor->setEnabled(true);
    ui->comboBoxCouseType->setEnabled(true);
}

void AdminDialog3::on_pushButtonSave_clicked()
{
    QDate beginDate=ui->dateEditBeginDate->date();
    QDate endDate=ui->dateEditEndDate->date();

    if(endDate < beginDate){
        QMessageBox::warning(this,"警告","开课时间晚于结课时间");
        return;
    }

    if(ui->lineEditCourseTerm->text().toInt() < 1 ){
        QMessageBox::warning(this,"警告","课程学期设置出错，请输入 1
或 输入 2");
        return;
    }
}

```

```

    }

    if(ui->lineEditHeadCount->text().toInt() < ui-
>lineEditSelectNum->text().toInt()){
        QMessageBox::warning(this,"警告","课程人数小于选课人数，设置
无效");
        return;
    }

    if(ui->lineEditCourseScore->text().toDouble()==0){
        QMessageBox::warning(this,"警告","课程学分设置无效");
        return;
    }

    QSqlRecord record = csql.findSingleRecord(courseId);
    bool flag = record.isNull("courseId");
    record.setValue("teacherId", ui->lineEditTeacherId-
>text().toInt());
    record.setValue("teacherName", ui->lineEditTeacherName-
>text());
    record.setValue("courseId", ui->lineEditCourseId-
>text().toInt());
    record.setValue("courseName", ui->lineEditCourseName->text());
    record.setValue("courseTerm", ui->lineEditCourseTerm-
>text().toInt());
    record.setValue("courseType", ui->comboBoxCouseType-
>currentIndex());
    record.setValue("courseScore", ui->lineEditCourseScore-
>text().toDouble());
    record.setValue("majorId", ui->comboBoxMajor->itemData(ui-
>comboBoxMajor->currentIndex()).toInt());
    record.setValue("courseBeginDate",beginDate.toString("yyyy-MM-
dd"));
    record.setValue("courseEndDate",endDate.toString("yyyy-MM-
dd"));
    record.setValue("courseClassTime", ui->lineEditClassTime-
>text());
    record.setValue("placeOfClass", ui->lineEditClassLocal-
>text());
    record.setValue("courseMaxCount", ui->lineEditHeadCount-
>text().toInt());
    record.setValue("courseSelectNum", ui->lineEditSelectNum-
>text().toInt());
    record.setValue("courseHour", ui->lineEditCourseHour-

```

```

>text().toInt());
    record.setValue("departmentId", ui->comboBoxDepartment-
>currentIndex()+1);

    if (flag) { // 添加新账户
        csql.insertSingleRecord(record);
    }
    else { // 更新账户信息
        csql.updateSingleRecord(record);
    }
    model->select();
    this->close();
//    QMessageBox::information(this, "提示", "已保存");
}

void AdminDialog3::on_comboBoxDepartment_currentIndexChanged(int
index)
{
    int dapartId=index+1;

    ui->comboBoxMajor->clear();

    MajorSql major;

    QSqlTableModel *majorModel=major.getTableModel(this);
    for(int i=0;i<majorModel->rowCount();++i)
    {
        QModelIndex departIndex=majorModel->index(i,majorModel-
>fieldIndex("departmentId"));
        if(majorModel-
>data(departIndex,Qt::EditRole).toInt()==dapartId)
        {
            QModelIndex index=majorModel->index(i,majorModel-
>fieldIndex("majorName"));
            QString name = majorModel-
>data(index,Qt::EditRole).toString();

            ui->comboBoxMajor->addItem(name,i+1);
        }
    }
}

```

```

void AdminDialog3::on_lineEditTeacherId_returnPressed()
{
    QString idName=ui->lineEditTeacherId->text();
    if(mp.count(idName)>0)
    {
        ui->lineEditTeacherName->setText(mp.value(idName));
    }
}

```

```

C++
//      课程管理对话框 ui 设计
#ifndef UI_ADMINDIALOG3_H
#define UI_ADMINDIALOG3_H

#include <QtCore/QVariant>
#include <QtWidgets/QApplication>
#include <QtWidgets/QComboBox>
#include <QtWidgets/QDateEdit>
#include <QtWidgets/QDialog>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QHBoxLayout>
#include <QtWidgets/QLabel>
#include <QtWidgets/QLineEdit>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QSpacerItem>
#include <QtWidgets/QVBoxLayout>

QT_BEGIN_NAMESPACE

class Ui_AdminDialog3
{
public:
    QGridLayout *gridLayout;
    QHBoxLayout *horizontalLayout_18;
    QVBoxLayout *verticalLayout;
    QHBoxLayout *horizontalLayout_3;
    QLabel *labelTeacherId;
    QLineEdit *lineEditTeacherId;
    QHBoxLayout *horizontalLayout_2;
    QLabel *labelCourseId;
    QLineEdit *lineEditCourseId;
    QHBoxLayout *horizontalLayout_4;

```

```
QLabel *labelCourseTerm;
QLineEdit *lineEditCourseTerm;
QHBoxLayout *horizontalLayout_6;
QLabel *labelCourseScore;
QLineEdit *lineEditCourseScore;
QHBoxLayout *horizontalLayout_7;
QLabel *labelBeginDate;
QDateEdit *dateEditBeginDate;
QHBoxLayout *horizontalLayout_11;
QLabel *labelClassTime;
QLineEdit *lineEditClassTime;
QHBoxLayout *horizontalLayout_13;
QLabel *labelHeadCount;
QLineEdit *lineEditHeadCount;
QHBoxLayout *horizontalLayout_12;
QLabel *labelMajorId;
QComboBox *comboBoxMajor;
QVBoxLayout *verticalLayout_2;
QHBoxLayout *horizontalLayout_9;
QLabel *labelTeacherName;
QLineEdit *lineEditTeacherName;
QHBoxLayout *horizontalLayout_10;
QLabel *labelCourseName;
QLineEdit *lineEditCourseName;
QHBoxLayout *horizontalLayout_5;
QLabel *labelCourseType;
QComboBox *comboBoxCourseType;
QHBoxLayout *horizontalLayout;
QLabel *labelCourseHour;
QLineEdit *lineEditCourseHour;
QHBoxLayout *horizontalLayout_8;
QLabel *labelEndDate;
QDateEdit *dateEditEndDate;
QHBoxLayout *horizontalLayout_14;
QLabel *labelClassLocal;
QLineEdit *lineEditClassLocal;
QHBoxLayout *horizontalLayout_17;
QLabel *labelSelectNum;
QLineEdit *lineEditSelectNum;
QHBoxLayout *horizontalLayout_15;
QLabel *labelDepartment;
QComboBox *comboBoxDepartment;
QHBoxLayout *horizontalLayout_16;
QPushButton *pushButtonEdit;
```

```

QSpacerItem *horizontalSpacer;
QPushButton *pushButtonSave;
QPushButton *pushButtonCancle;

void setupUi(QDialog *AdminDialog3)
{
    if (AdminDialog3->objectName().isEmpty())
        AdminDialog3->setObjectName("AdminDialog3");
    AdminDialog3->resize(599, 378);
    AdminDialog3->setMaximumSize(QSize(599, 378));
    gridLayout = new QGridLayout(AdminDialog3);
    gridLayout->setObjectName("gridLayout");
    horizontalLayout_18 = new QHBoxLayout();
    horizontalLayout_18->setObjectName("horizontalLayout_18");
    verticalLayout = new QVBoxLayout();
    verticalLayout->setObjectName("verticalLayout");
    horizontalLayout_3 = new QHBoxLayout();
    horizontalLayout_3->setObjectName("horizontalLayout_3");
    labelTeacherId = new QLabel(AdminDialog3);
    labelTeacherId->setObjectName("labelTeacherId");

    horizontalLayout_3->addWidget(labelTeacherId);

    lineEditTeacherId = new QLineEdit(AdminDialog3);
    lineEditTeacherId->setObjectName("lineEditTeacherId");
    lineEditTeacherId->setMinimumSize(QSize(0, 30));

    horizontalLayout_3->addWidget(lineEditTeacherId);

    verticalLayout->addLayout(horizontalLayout_3);

    horizontalLayout_2 = new QHBoxLayout();
    horizontalLayout_2->setObjectName("horizontalLayout_2");
    labelCourseId = new QLabel(AdminDialog3);
    labelCourseId->setObjectName("labelCourseId");

    horizontalLayout_2->addWidget(labelCourseId);

    lineEditCourseId = new QLineEdit(AdminDialog3);
    lineEditCourseId->setObjectName("lineEditCourseId");
    lineEditCourseId->setMinimumSize(QSize(0, 30));

    horizontalLayout_2->addWidget(lineEditCourseId);

```

```
verticalLayout->addLayout(horizontalLayout_2);

horizontalLayout_4 = new QHBoxLayout();
horizontalLayout_4->setObjectName("horizontalLayout_4");
labelCourseTerm = new QLabel(AdminDialog3);
labelCourseTerm->setObjectName("labelCourseTerm");

horizontalLayout_4->addWidget(labelCourseTerm);

lineEditCourseTerm = new QLineEdit(AdminDialog3);
lineEditCourseTerm->setObjectName("lineEditCourseTerm");
lineEditCourseTerm->setMinimumSize(QSize(0, 30));

horizontalLayout_4->addWidget(lineEditCourseTerm);

verticalLayout->addLayout(horizontalLayout_4);

horizontalLayout_6 = new QHBoxLayout();
horizontalLayout_6->setObjectName("horizontalLayout_6");
labelCourseScore = new QLabel(AdminDialog3);
labelCourseScore->setObjectName("labelCourseScore");

horizontalLayout_6->addWidget(labelCourseScore);

lineEditCourseScore = new QLineEdit(AdminDialog3);
lineEditCourseScore->setObjectName("lineEditCourseScore");
lineEditCourseScore->setMinimumSize(QSize(0, 30));

horizontalLayout_6->addWidget(lineEditCourseScore);

verticalLayout->addLayout(horizontalLayout_6);

horizontalLayout_7 = new QHBoxLayout();
horizontalLayout_7->setObjectName("horizontalLayout_7");
labelBeginDate = new QLabel(AdminDialog3);
labelBeginDate->setObjectName("labelBeginDate");
labelBeginDate->setMinimumSize(QSize(60, 0));
labelBeginDate->setMaximumSize(QSize(47, 16777215));

horizontalLayout_7->addWidget(labelBeginDate);
```



```
dateEditBeginDate = new QDateEdit(AdminDialog3);
dateEditBeginDate->setObjectName("dateEditBeginDate");
dateEditBeginDate->setMinimumSize(QSize(0, 30));

horizontalLayout_7->addWidget(dateEditBeginDate);

verticalLayout->addLayout(horizontalLayout_7);

horizontalLayout_11 = new QHBoxLayout();
horizontalLayout_11->setObjectName("horizontalLayout_11");
labelClassTime = new QLabel(AdminDialog3);
labelClassTime->setObjectName("labelClassTime");

horizontalLayout_11->addWidget(labelClassTime);

lineEditClassTime = new QLineEdit(AdminDialog3);
lineEditClassTime->setObjectName("lineEditClassTime");
lineEditClassTime->setMinimumSize(QSize(0, 30));

horizontalLayout_11->addWidget(lineEditClassTime);

verticalLayout->addLayout(horizontalLayout_11);

horizontalLayout_13 = new QHBoxLayout();
horizontalLayout_13->setObjectName("horizontalLayout_13");
labelHeadCount = new QLabel(AdminDialog3);
labelHeadCount->setObjectName("labelHeadCount");

horizontalLayout_13->addWidget(labelHeadCount);

lineEditHeadCount = new QLineEdit(AdminDialog3);
lineEditHeadCount->setObjectName("lineEditHeadCount");
lineEditHeadCount->setMinimumSize(QSize(0, 30));

horizontalLayout_13->addWidget(lineEditHeadCount);

verticalLayout->addLayout(horizontalLayout_13);

horizontalLayout_12 = new QHBoxLayout();
horizontalLayout_12->setObjectName("horizontalLayout_12");
```

```
labelMajorId = new QLabel(AdminDialog3);
labelMajorId->setObjectName("labelMajorId");

horizontalLayout_12->addWidget(labelMajorId);

comboBoxMajor = new QComboBox(AdminDialog3);
comboBoxMajor->setObjectName("comboBoxMajor");
comboBoxMajor->setMinimumSize(QSize(101, 24));

horizontalLayout_12->addWidget(comboBoxMajor);

verticalLayout->addLayout(horizontalLayout_12);

horizontalLayout_18->addLayout(verticalLayout);

verticalLayout_2 = new QVBoxLayout();
verticalLayout_2->setObjectName("verticalLayout_2");
horizontalLayout_9 = new QHBoxLayout();
horizontalLayout_9->setObjectName("horizontalLayout_9");
labelTeacherName = new QLabel(AdminDialog3);
labelTeacherName->setObjectName("labelTeacherName");

horizontalLayout_9->addWidget(labelTeacherName);

lineEditTeacherName = new QLineEdit(AdminDialog3);
lineEditTeacherName->setObjectName("lineEditTeacherName");
lineEditTeacherName->setMinimumSize(QSize(0, 30));

horizontalLayout_9->addWidget(lineEditTeacherName);

verticalLayout_2->addLayout(horizontalLayout_9);

horizontalLayout_10 = new QHBoxLayout();
horizontalLayout_10->setObjectName("horizontalLayout_10");
labelCourseName = new QLabel(AdminDialog3);
labelCourseName->setObjectName("labelCourseName");

horizontalLayout_10->addWidget(labelCourseName);

lineEditCourseName = new QLineEdit(AdminDialog3);
lineEditCourseName->setObjectName("lineEditCourseName");
```

```
lineEditCourseName->setMinimumSize(QSize(0, 30));

horizontalLayout_10->addWidget(lineEditCourseName);


verticalLayout_2->addLayout(horizontalLayout_10);


horizontalLayout_5 = new QHBoxLayout();
horizontalLayout_5->setObjectName("horizontalLayout_5");
labelCourseType = new QLabel(AdminDialog3);
labelCourseType->setObjectName("labelCourseType");
labelCourseType->setMinimumSize(QSize(60, 0));
labelCourseType->setMaximumSize(QSize(47, 16777215));

horizontalLayout_5->addWidget(labelCourseType);


comboBoxCourseType = new QComboBox(AdminDialog3);
comboBoxCourseType->setObjectName("comboBoxCourseType");
comboBoxCourseType->setMinimumSize(QSize(70, 30));

horizontalLayout_5->addWidget(comboBoxCourseType);


verticalLayout_2->addLayout(horizontalLayout_5);


horizontalLayout = new QHBoxLayout();
horizontalLayout->setObjectName("horizontalLayout");
labelCourseHour = new QLabel(AdminDialog3);
labelCourseHour->setObjectName("labelCourseHour");
labelCourseHour->setMinimumSize(QSize(60, 0));

horizontalLayout->addWidget(labelCourseHour);


lineEditCourseHour = new QLineEdit(AdminDialog3);
lineEditCourseHour->setObjectName("lineEditCourseHour");
lineEditCourseHour->setMinimumSize(QSize(0, 30));

horizontalLayout->addWidget(lineEditCourseHour);


verticalLayout_2->addLayout(horizontalLayout);


horizontalLayout_8 = new QHBoxLayout();
horizontalLayout_8->setObjectName("horizontalLayout_8");
```

```
labelEndDate = new QLabel(AdminDialog3);
labelEndDate->setObjectName("labelEndDate");
labelEndDate->setMinimumSize(QSize(60, 0));
labelEndDate->setMaximumSize(QSize(47, 16777215));

horizontalLayout_8->addWidget(labelEndDate);

dateEditEndDate = new QDateEdit(AdminDialog3);
dateEditEndDate->setObjectName("dateEditEndDate");
dateEditEndDate->setMinimumSize(QSize(0, 30));

horizontalLayout_8->addWidget(dateEditEndDate);

verticalLayout_2->addLayout(horizontalLayout_8);

horizontalLayout_14 = new QHBoxLayout();
horizontalLayout_14->setObjectName("horizontalLayout_14");
labelClassLocal = new QLabel(AdminDialog3);
labelClassLocal->setObjectName("labelClassLocal");

horizontalLayout_14->addWidget(labelClassLocal);

lineEditClassLocal = new QLineEdit(AdminDialog3);
lineEditClassLocal->setObjectName("lineEditClassLocal");
lineEditClassLocal->setMinimumSize(QSize(0, 30));

horizontalLayout_14->addWidget(lineEditClassLocal);

verticalLayout_2->addLayout(horizontalLayout_14);

horizontalLayout_17 = new QHBoxLayout();
horizontalLayout_17->setObjectName("horizontalLayout_17");
labelSelectNum = new QLabel(AdminDialog3);
labelSelectNum->setObjectName("labelSelectNum");

horizontalLayout_17->addWidget(labelSelectNum);

lineEditSelectNum = new QLineEdit(AdminDialog3);
lineEditSelectNum->setObjectName("lineEditSelectNum");
lineEditSelectNum->setMinimumSize(QSize(0, 30));

horizontalLayout_17->addWidget(lineEditSelectNum);
```

```
verticalLayout_2->addLayout(horizontalLayout_17);

horizontalLayout_15 = new QHBoxLayout();
horizontalLayout_15->setObjectName("horizontalLayout_15");
labelDepartment = new QLabel(AdminDialog3);
labelDepartment->setObjectName("labelDepartment");

horizontalLayout_15->addWidget(labelDepartment);

comboBoxDepartment = new QComboBox(AdminDialog3);
comboBoxDepartment->setObjectName("comboBoxDepartment");
comboBoxDepartment->setMinimumSize(QSize(101, 24));

horizontalLayout_15->addWidget(comboBoxDepartment);

verticalLayout_2->addLayout(horizontalLayout_15);

horizontalLayout_18->addLayout(verticalLayout_2);

gridLayout->addLayout(horizontalLayout_18, 0, 0, 1, 1);

horizontalLayout_16 = new QHBoxLayout();
horizontalLayout_16->setObjectName("horizontalLayout_16");
pushButtonEdit = new QPushButton(AdminDialog3);
pushButtonEdit->setObjectName("pushButtonEdit");

horizontalLayout_16->addWidget(pushButtonEdit);

horizontalSpacer = new QSpacerItem(81, 20,
QSizePolicy::Expanding, QSizePolicy::Minimum);

horizontalLayout_16->addItem(horizontalSpacer);

pushButtonSave = new QPushButton(AdminDialog3);
pushButtonSave->setObjectName("pushButtonSave");

horizontalLayout_16->addWidget(pushButtonSave);

pushButtonCancle = new QPushButton(AdminDialog3);
```

```

        pushButtonCancle->setObjectName("pushButtonCancle");

        horizontalLayout_16->addWidget(pushButtonCancle);

        gridLayout->addLayout(horizontalLayout_16, 1, 0, 1, 1);

        retranslateUi(AdminDialog3);
        QObject::connect(pushButtonCancle, &QPushButton::clicked,
AdminDialog3, qOverload<>(&QDialog::close));

        QMetaObject::connectSlotsByName(AdminDialog3);
    } // setupUi

    void retranslateUi(QDialog *AdminDialog3)
    {
        AdminDialog3-
>setWindowTitle(QCoreApplication::translate("AdminDialog3",
"\350\257\276\347\250\213\347\256\241\347\220\206\345\257\271\350\
257\235\346\241\206", nullptr));
        labelTeacherId-
>setText(QCoreApplication::translate("AdminDialog3",
"\346\225\231\345\270\210\347\274\226\345\217\267", nullptr));
        labelCourseId-
>setText(QCoreApplication::translate("AdminDialog3",
"\350\257\276\347\250\213\347\274\226\345\217\267", nullptr));
        labelCourseTerm-
>setText(QCoreApplication::translate("AdminDialog3",
"\350\257\276\347\250\213\345\255\246\346\234\237", nullptr));
        labelCourseScore-
>setText(QCoreApplication::translate("AdminDialog3",
"\350\257\276\347\250\213\345\255\246\345\210\206", nullptr));
        labelBeginDate-
>setText(QCoreApplication::translate("AdminDialog3",
"\345\274\200\350\257\276\346\227\266\351\227\264", nullptr));
        labelClassTime-
>setText(QCoreApplication::translate("AdminDialog3",
"\344\270\212\350\257\276\346\227\266\351\227\264", nullptr));
        labelHeadCount-
>setText(QCoreApplication::translate("AdminDialog3",
"\350\257\276\347\250\213\344\272\272\346\225\260", nullptr));
        labelMajorId-
>setText(QCoreApplication::translate("AdminDialog3", "\344\270\223

```

```

\344\270\232", nullptr));
    labelTeacherName-
>setText(QCoreApplication::translate("AdminDialog3",
"\346\225\231\345\270\210\345\247\223\345\220\215", nullptr));
    labelCourseName-
>setText(QCoreApplication::translate("AdminDialog3",
"\350\257\276\347\250\213\345\220\215\347\247\260", nullptr));
    labelCourseType-
>setText(QCoreApplication::translate("AdminDialog3",
"\350\257\276\347\250\213\347\261\273\345\236\213", nullptr));
    labelCourseHour-
>setText(QCoreApplication::translate("AdminDialog3", "\350\257\276
\346\227\266", nullptr));
    labelEndDate-
>setText(QCoreApplication::translate("AdminDialog3",
"\347\273\223\350\257\276\346\227\266\351\227\264", nullptr));
    labelClassLocal-
>setText(QCoreApplication::translate("AdminDialog3",
"\344\270\212\350\257\276\345\234\260\347\202\271", nullptr));
    labelSelectNum-
>setText(QCoreApplication::translate("AdminDialog3",
"\345\267\262\351\200\211\344\272\272\346\225\260", nullptr));
    labelDepartment-
>setText(QCoreApplication::translate("AdminDialog3", "\351\231\242
\347\263\273", nullptr));
    pushButtonEdit-
>setText(QCoreApplication::translate("AdminDialog3",
"\347\274\226\350\276\221", nullptr));
    pushButtonSave-
>setText(QCoreApplication::translate("AdminDialog3",
"\344\277\235\345\255\230", nullptr));
    pushButtonCancle-
>setText(QCoreApplication::translate("AdminDialog3",
"\345\217\226\346\266\210", nullptr));
    } // retranslateUi

};

namespace Ui {
    class AdminDialog3: public Ui_AdminDialog3 {};
} // namespace Ui

QT_END_NAMESPACE

```

```
#endif // UI_ADMINDIALOG3_H
```

四、系统测试

首先是功能测试。

管理员登录，然后个人账户设置功能测试，对账户的修改保存操作进行测试。对账户的年龄、性别输入任意字符任意数字，点击保存如果输入不符合相应格式（性别为男|女，年龄在 15-140 岁），应当提示输入不正确。

教师管理功能测试。添加教师，测试胡乱输入有没有相应警报提示输入出错无法保存，或者直接无法胡乱输入。以同样的方式测试教师管理模块的修改，删除功能是否可以正常使用，或者非正常修改提示报警。

学生管理功能测试，课程管理功能测试方法与教师管理模块一致。

选课功能测试，通过选择学生或淘汰学生，登录相应学生账户查看选课成功或失败，同时教师学生选课也应发生相应改变。

评教功能测试，判断展示的评教记录是否与学生评教结果一致，测试删除功能

成绩管理测试，检查所展示的成绩分数是否与教师所打分一致，测试自动评判功能（60 及以上通过，60 分以下不能通过），提交功能是否正常运作。

公告管理功能测试，通过增加删除公告，然后登录学生端，教师端查看展示公告是否与管理员修改后的一致。

经过组内三个成员多次输入测试，所有模块都能正常运行。教师端、学生端测试与管理员端测试方法一致，此处不再赘述。

其次是整体测试，组内成员分别扮演管理员，学生，老师三种身份：

- 1、先由管理员组员登录管理员账户，然后直接修改登录信息和密码，退出再重新登录测试修改是否成功。成功后管理员组员创建两个老师，两个学生，同时为每个老师创建四门课程，然后退出。

- 2、由学生组员选择其中一个学生账号登录，登录成功后修改账户信息和密码，退出再重新登录测试修改是否成功。成功后开始选课，然后退出登录，对另一个学生账号执行相同的操作，最后退出登录。

- 3、老师组员选择其中一个老师账号登录，登录成功后修改账户信息和密码，退出再重新登录测试修改是否成功。成功后开始在选课列表筛选在选课期间选择了自己所教授课程的学生，完成后给为筛选通过的其中一个学生进行成绩打分，最后退出登录。

4、管理员组员登录之前的账号，点击选课界面查看当前所有学生的选课状态，淘汰为打分的学生。然后进入成绩管理界面，点击自动判断按钮，判断学生成绩是否通过（60 分及以上）。如果未通过，自动更改补考状态为 1，然后退出登录。

5、学生组员登录之前的账号，查看选课是否成功，已经被打分的学生查看成绩单，然后进入评教端，对已经打分并且审核后的课程老师进行评教，最后退出程序。

6、老师组员登录之前的账号，点击评教窗口查看评教分数和分析图例。

7、管理员组员再次登录，增加一门公告，删除一门原有公告，修改一门原有公告，然后退出。学生组员，老师组员登录后，查看公告。

按以上次序执行三遍，并未发现错误，整体测试通过，系统测试完成。

五、项目展望

在该项目的开发中，我们已经解决了教务管理系统所必须拥有的基本功能，包括选退课管理、公告管理、学生，教师管理、成绩管理、课程管理等；解决了学生、教师、管理员之间数据传递、增加、修改、删除的相互依赖问题，基本做到了一个教务系统所必须的基本素质，但由于时间、技术上的欠缺，仍有以下问题我们尚未解决。

- 1、没有为学生设计班级属性，没有完整的课程表展示。
- 2、没有课程资源管理，请假管理，调课管理等。
- 3、没有额外的功能拓展（如图书查询，培养大纲查询等）。

同时由于技术能力和开发时间不足，该项目算法的时间复杂度较高，代码较为冗余等问题尚未解决。之后我们将努力学习新技术，掌握更高明的算法，对该项目优化并进行以下方面的拓展：

1. 增加课程排课、课程评价、课程资源管理等功能，使教师和学生能够更好地管理和参与课程学习。
2. 开发移动端应用，使学生、教师和家长能够通过手机随时随地查看课程表、成绩、通知等信息。
3. 加强系统的安全性，包括用户身份验证、数据加密、访问控制等，保护学校和学生的隐私和安全。
4. 实现教务管理系统与图书馆管理系统的集成，使学生能够方便地查询图书馆藏书和借阅情况。

争取实现功能更完善，性能更优秀，可以实际使用的教务系统。

